BANNARI AMMAN INSTITUTE OF TECHNOLOGY
An Autonomous Institution Affiliated to Anna University Chennai - Approved by AICTE - Accredited by NAAC with "A" Grade
SATHYAMANGALAM - 638 401 ERODE DISTRICT TAMIL NADU INDIA
Ph: 04295-226000 / 221289 Fax: 04295-226666 E-mail: stayahead@bitsathy.ac.in Web: www.bitsathy.ac.in

ONE CREDIT ASSIGNMENT REPORT

ONE CREDIT COURSE ASSIGNMENT REPORT

Submitted by
ANOOSKAVIN G
(191CS119)
Final year - CSE

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

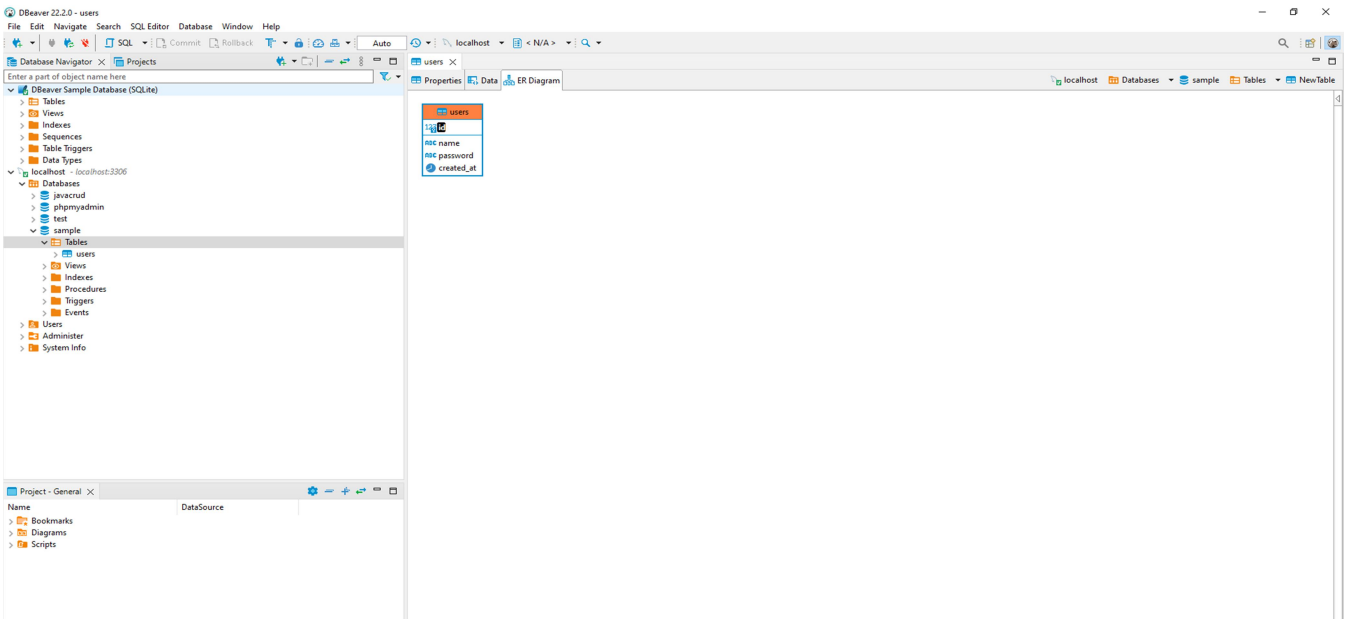
(An Autonomous Institution Affiliated to Anna University, Chennai)
SATHYAMANGALAM-638401

September 2022

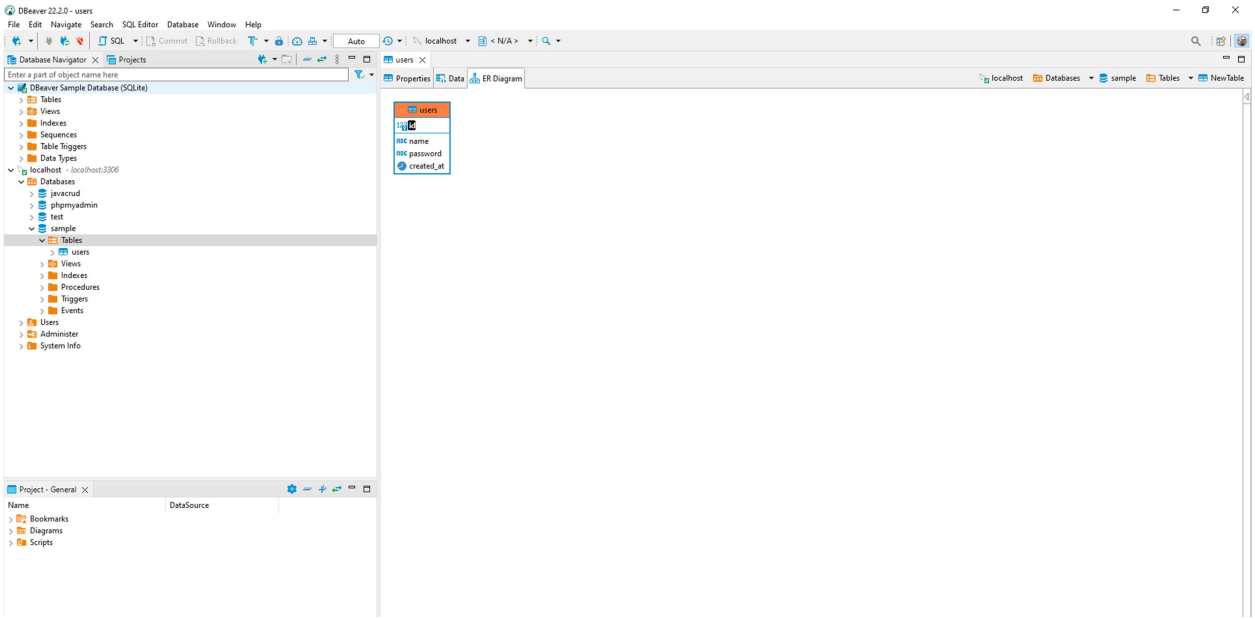
Ex.no: 1	Create a Table with 4 columns with primary key and time stamp data-type
Date: 24.09.2022	

Implementation Steps & Output:

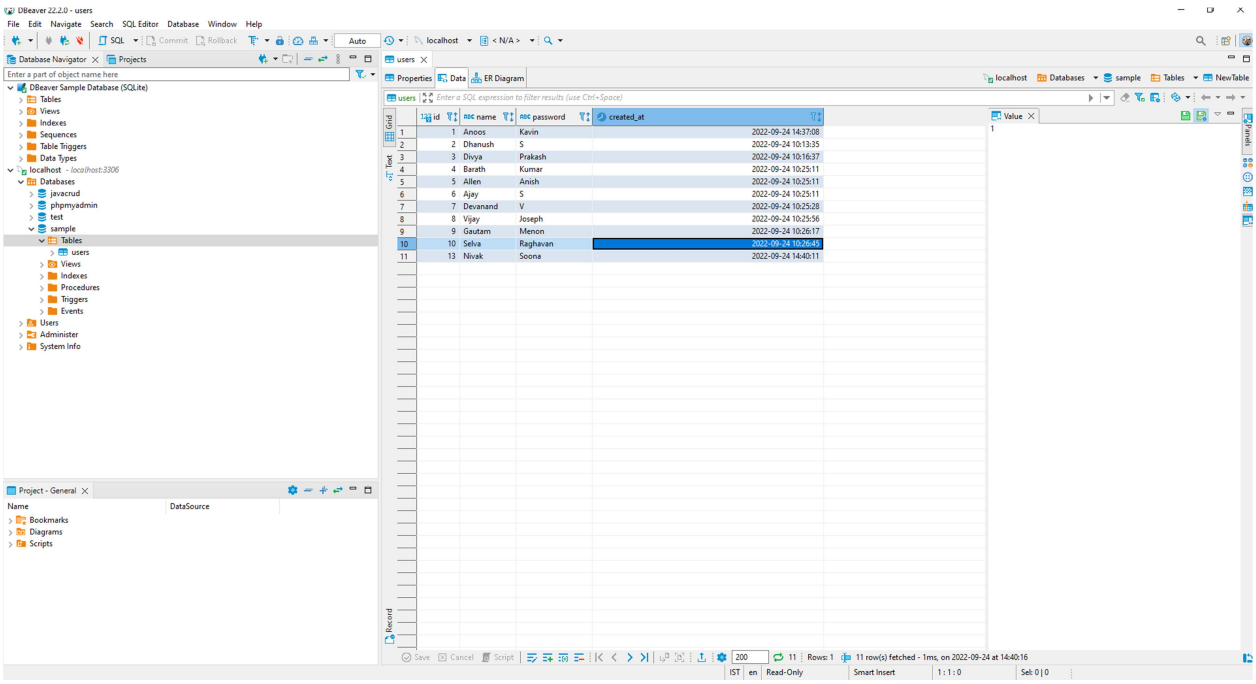
1. Open **DBeaver.** Create **MySql** database in our DBeaver.
2. Put the credentials for DB and connect it.
3. Create a table by right clicking the database and name it.
4. Create the columns for the table you created.



5. Enter the constraints for the columns like **Primary key, Not null etc,.**



6. Add the values for the table.



We can also create the following table using **Queries** given below,

```
CREATE DATABASE sample;
CREATE TABLE users(
id  INT NOT NULL AUTO INCREMENT,
name VARCHAR(100) NOT NULL,
password VARCHAR(100) NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
PRIMARY KEY(id)
);
```

Ex.no: 2	Write All API methods with query using **POSTMAN**
Date: 24.09.2022	

Implementation Steps & Output:

1. Write the code for the required packages to use it such as **mysql,body-parser,cors,express**

```
var expres = require('express')
var app = expres()
var bodyparser = require('body-parser')
var mysql = require('mysql')
var cors = require('cors')

app.use(cors())
app.use(bodyparser.json())

app.use(bodyparser.urlencoded({
    extended : true
}));
```

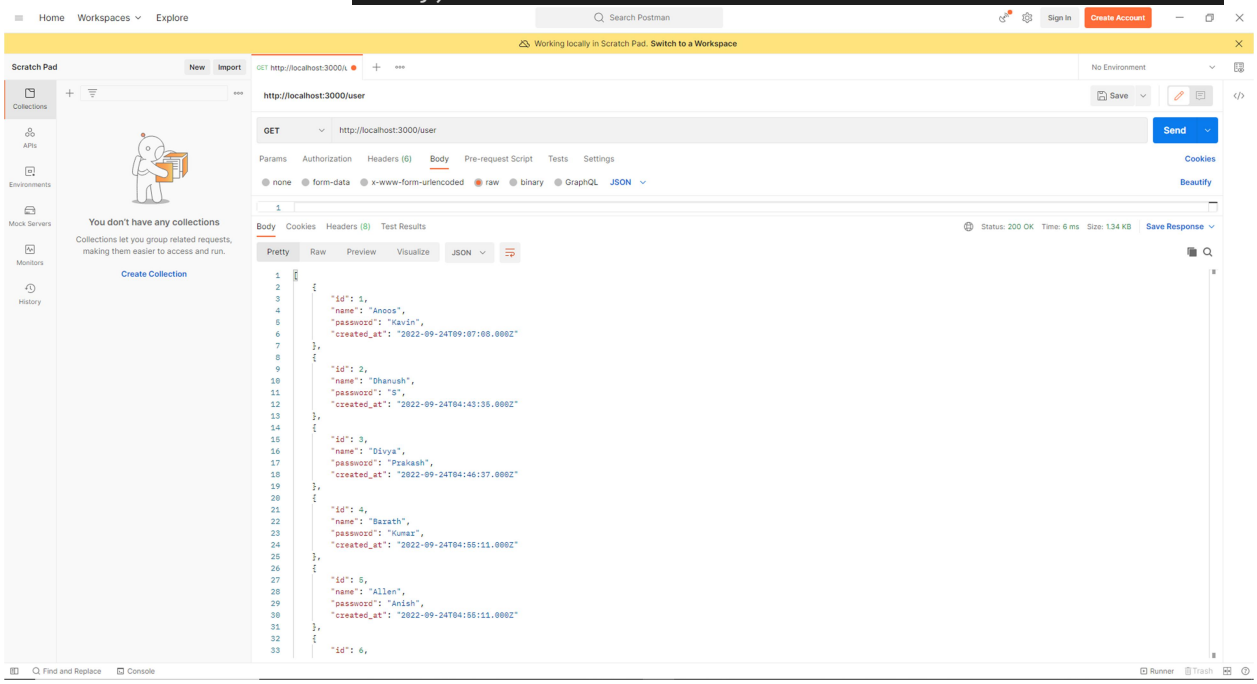
2. Code for connecting MySQL DB.

```
var dbConn = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: '',
    database:'sample'
})
dbConn.connect()
```

3. **GET METHOD:**
   - Obtain all the values of the table using GET method.

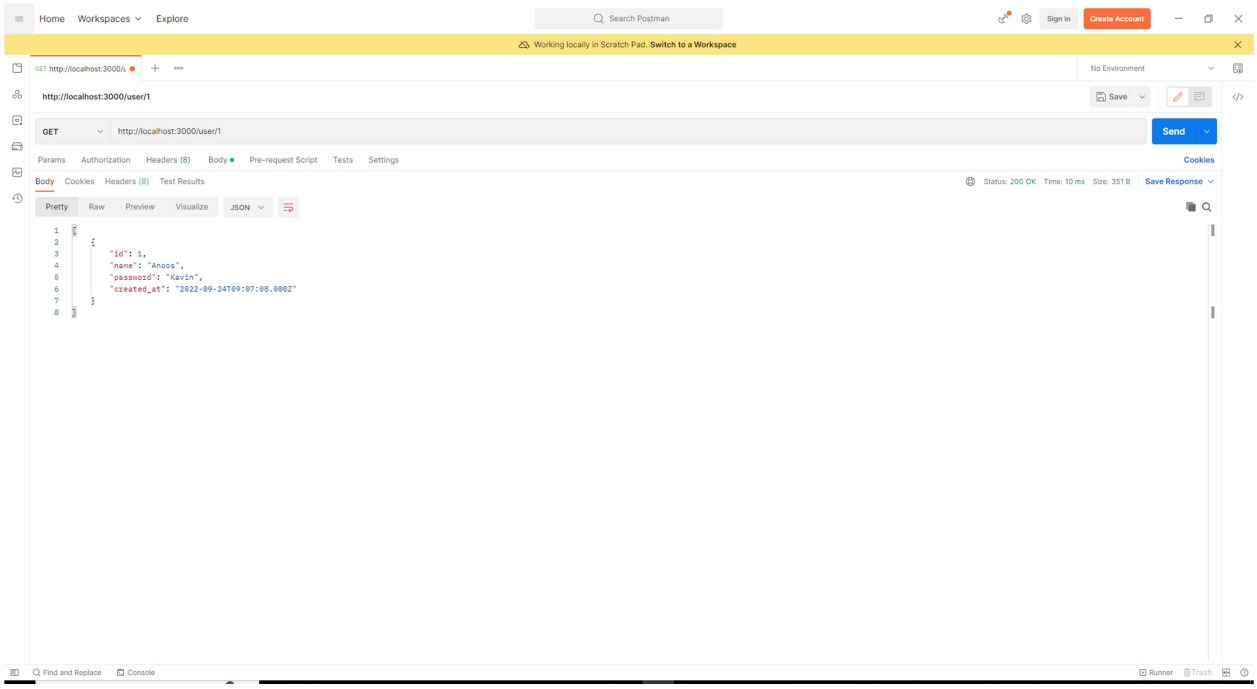
```
  app.get('/user',function(req,res){

  dbConn.query('select * from
users',function(error,results,fields){
        if(error) throw error;
        return res.send(results);
```

```
    })
})
```



- Obtain the details for the particular id using GET method

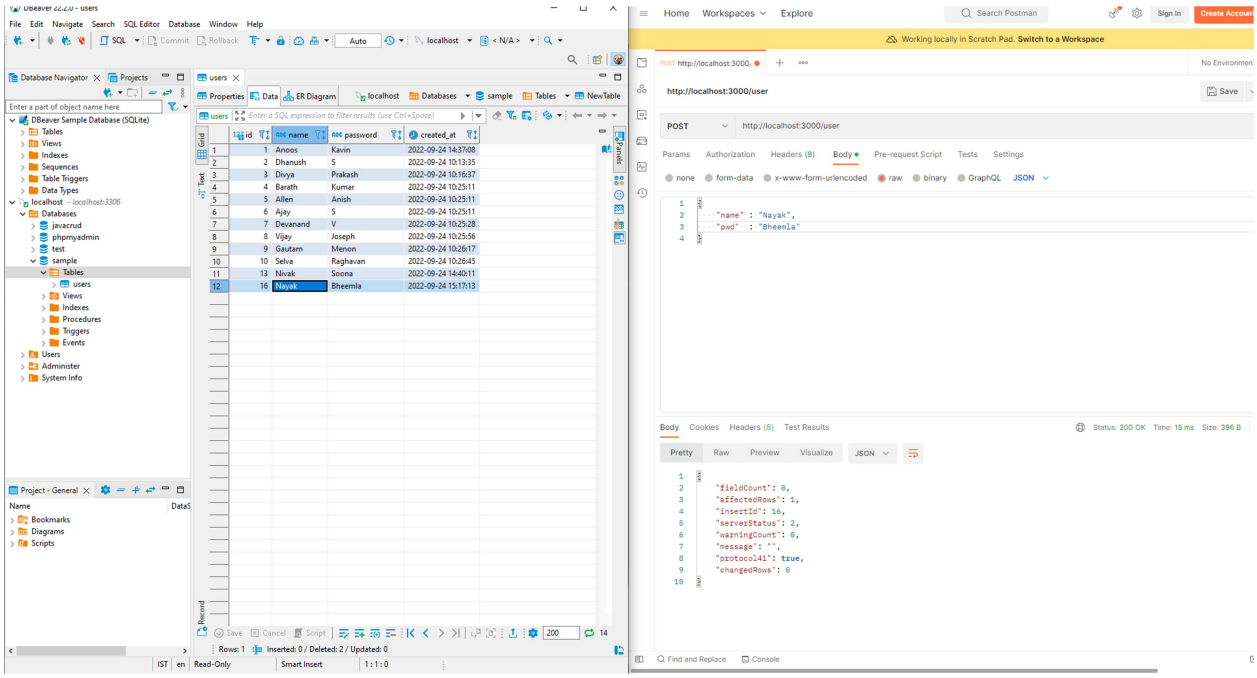
```
    app.get('/user/:id',function(req,res){

    let id = req.params.id;
    console.log(id);
    dbConn.query(`select * from users where id = ${id}
`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
    })
})
```



4. **POST METHOD:**

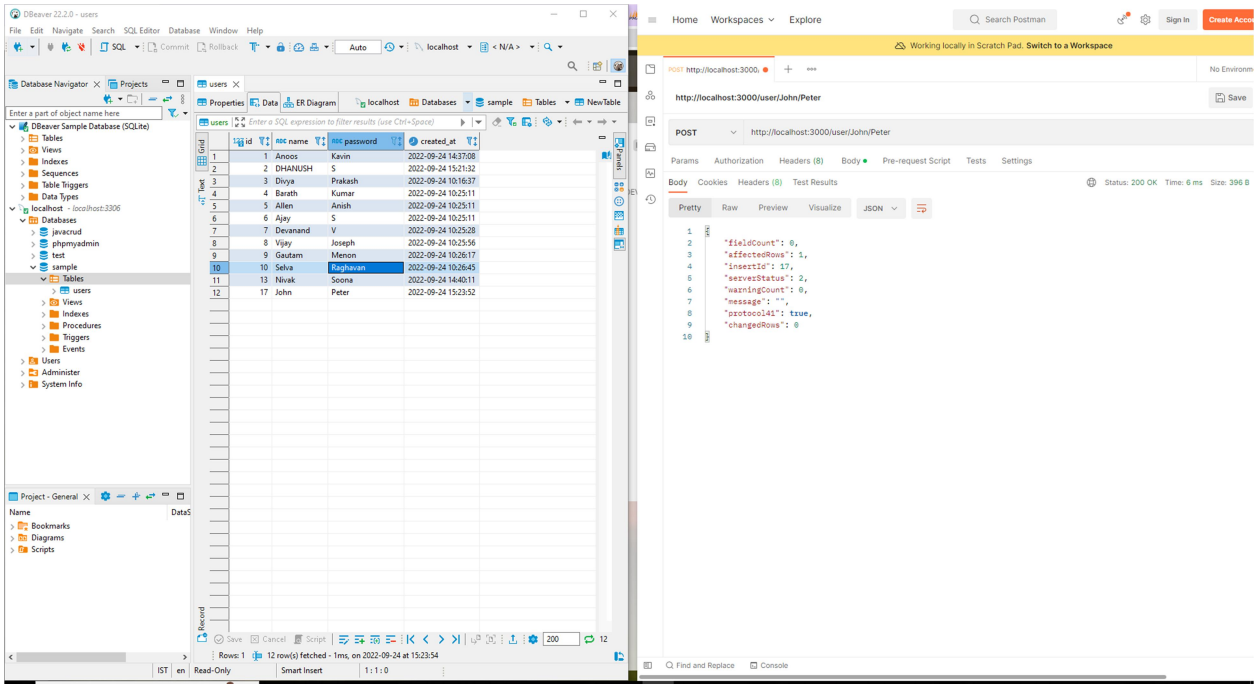
- Create a user by passing values via **body**

```
   app.post('/user',function(req,res){

    let name = req.body.name;
    let pwd = req.body.pwd;
    console.log(name,pwd);
    dbConn.query(`insert into users(name,password)
values('${name}','${pwd}')`,function(error,results,field
s){
        if(error) throw error;
        return res.send(results);
})
})
```



- Create a user by passing values via **Params**

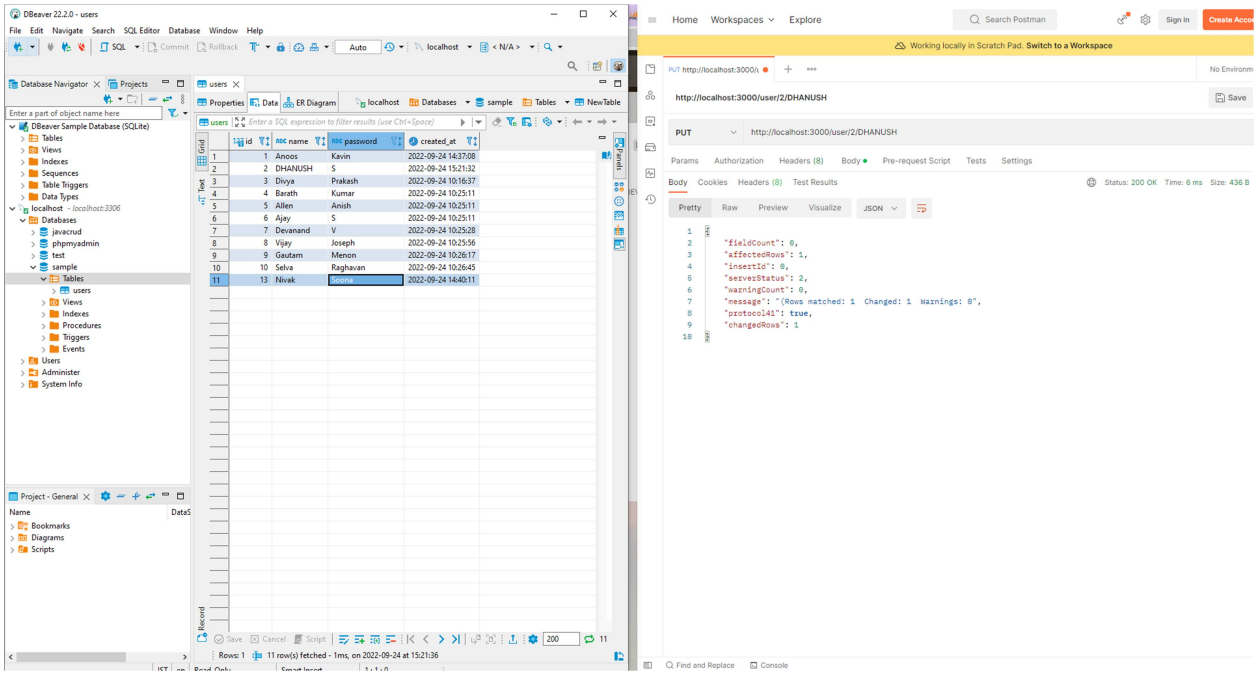
```
    app.post('/user/:name/:pwd',function(req,res){

    let name = req.params.name;
    let pwd = req.params.pwd;
    console.log(name,pwd);
    dbConn.query(`insert into users(name,password)
values('${name}','${pwd}')`,function(error,results,field
s){
        if(error) throw error;
        return res.send(results);
})
})
```



5. PUT METHOD

- Update the user name with respect to id by using PUT method.

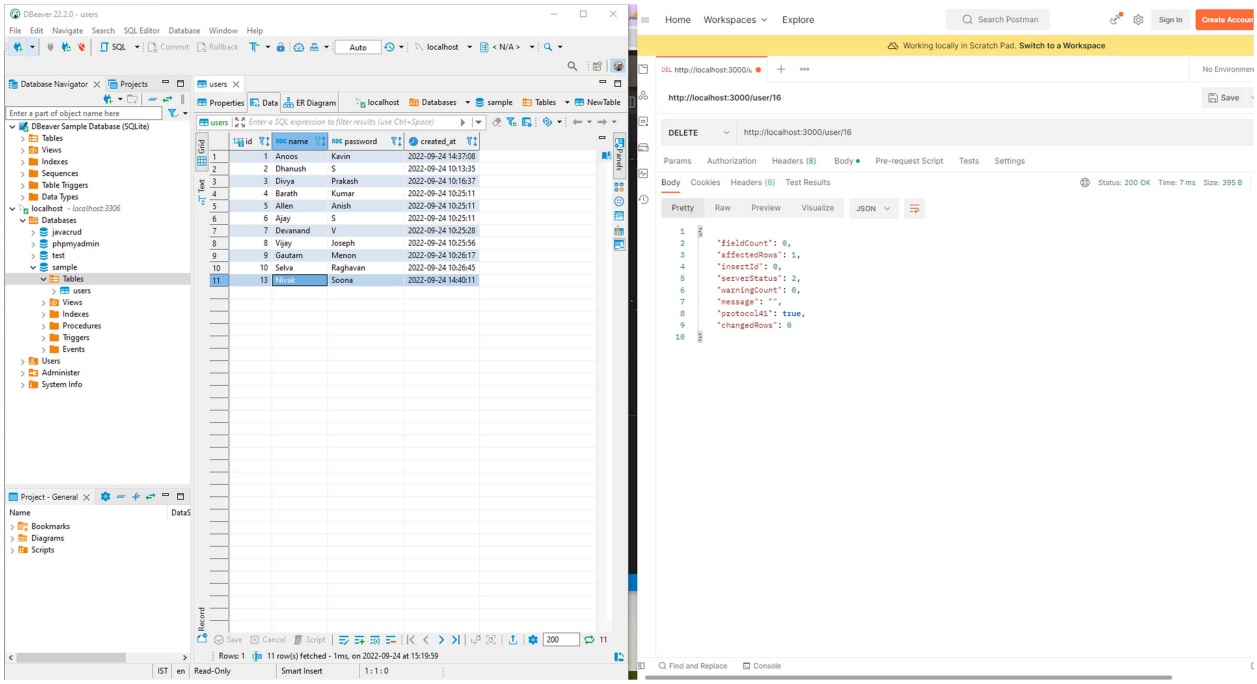
```
    app.put('/user/:id/:name',function(req,res){

    let id = req.params.id;
    let name = req.params.name;
    console.log(id);
    dbConn.query(`update  users set
name='${name}'  where id = ${id}
`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
    })
})
```



6. DELETE METHOD

- Delete a user with respect to id by using DELETE method.

```
app.delete('/user/:id',function(req,res){

    let id = req.params.id;
    console.log(id);
    dbConn.query(`delete  from users where id =
${id}`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
    })
})
```



Full Code:

```
var expres = require('express')

var app = expres()
var bodyparser = require('body-parser')
var mysql = require('mysql')
var cors = require('cors')

app.use(cors())
app.use(bodyparser.json())

app.use(bodyparser.urlencoded({
    extended : true
}));
var dbConn = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: '',
    database:'sample'
})
dbConn.connect()

app.get('/user',function(req,res){
    dbConn.query('select * from users',function(error,results,fields){
```

```
        if(error) throw error;
        return res.send(results);
    })
})

app.post('/user',function(req,res){
    let name = req.body.name;
    let pwd = req.body.pwd;
    console.log(name,pwd);
    dbConn.query(`insert into users(name,password)
values('${name}','${pwd}')`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
})
})

app.get('/user/:id',function(req,res){
    let id = req.params.id;
    console.log(id);
    dbConn.query(`select * from users where id = ${id}
`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
    })
})

app.delete('/user/:id',function(req,res){
    let id = req.params.id;
    console.log(id);
    dbConn.query(`delete  from users where id =
${id}`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
    })
})

app.put('/user/:id/:name',function(req,res){
    let id = req.params.id;
    let name = req.params.name;
    console.log(id);
    dbConn.query(`update  users set name='${name}'  where id = ${id}
`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
    })
```

```
})

app.post('/user/:name/:pwd',function(req,res){
    let name = req.params.name;
    let pwd = req.params.pwd;
    console.log(name,pwd);
    dbConn.query(`insert into users(name,password)
values('${name}','${pwd}')`,function(error,results,fields){
        if(error) throw error;
        return res.send(results);
})
})

app.listen(3000,function(){
    console.log("Server Started");
})
```